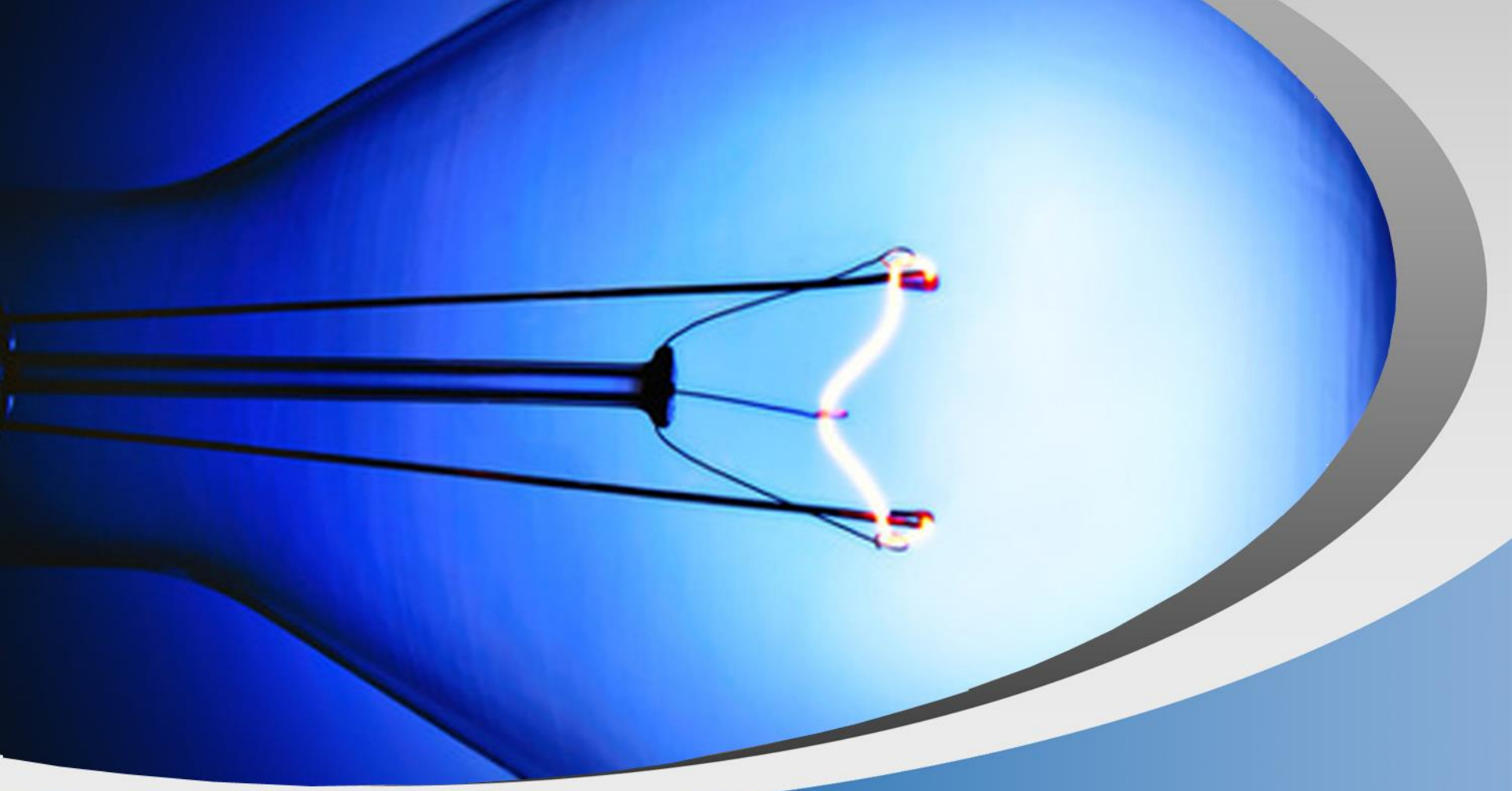


SystemC CCI WG
User Defined Data Type

Girish Verma, P V S Phaneendra, CircuitSutra Technologies Pvt. Ltd, September 2011

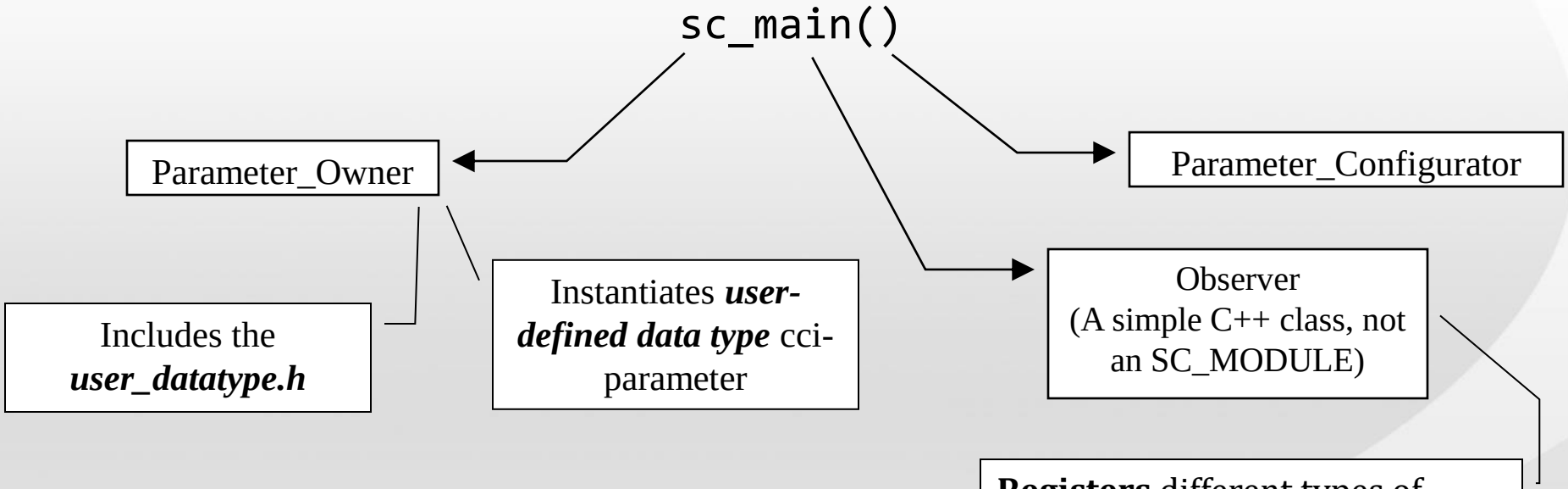
accellera
SYSTEMS INITIATIVE

User Defined Data Types

- **Objective of the present example is to demonstrate the following :**
  - Adding support for user-defined data types, including complex structures

- **Show access using both JSON values and cci_value**

User Defined Data Types



Note#1 : *For a detailed understanding of registration and implementation of callbacks, please go through the presentation for ex#8 or ex#11 and ex#12*

Within the Callbacks implementation :
Name of the originator : ***ev.originator.name();***

Note#2 : *The originator information will be displayed by the shape on the right*

Reporter : *Observer*
Callback Type : *cb_type*
Originator : *originator_name*

User Defined Data Types

Step#1 : ***Defining a User-Defined Data Type***

Declare a C++ class/structure for the *user-defined data type* (see the following slide)

Create a default constructor for the user-defined data type structure

Also provide an overloaded constructor that is useful to assign the user-defined values in the desired manner

Implement cci_value_converter<user-defined-type> with pack and unpack functions (not shown – see example code)

Overload the insertion (<<) operator to provide convenience in reporting

User Defined Data Types

Infrastructure to be created : Creating Data Structure

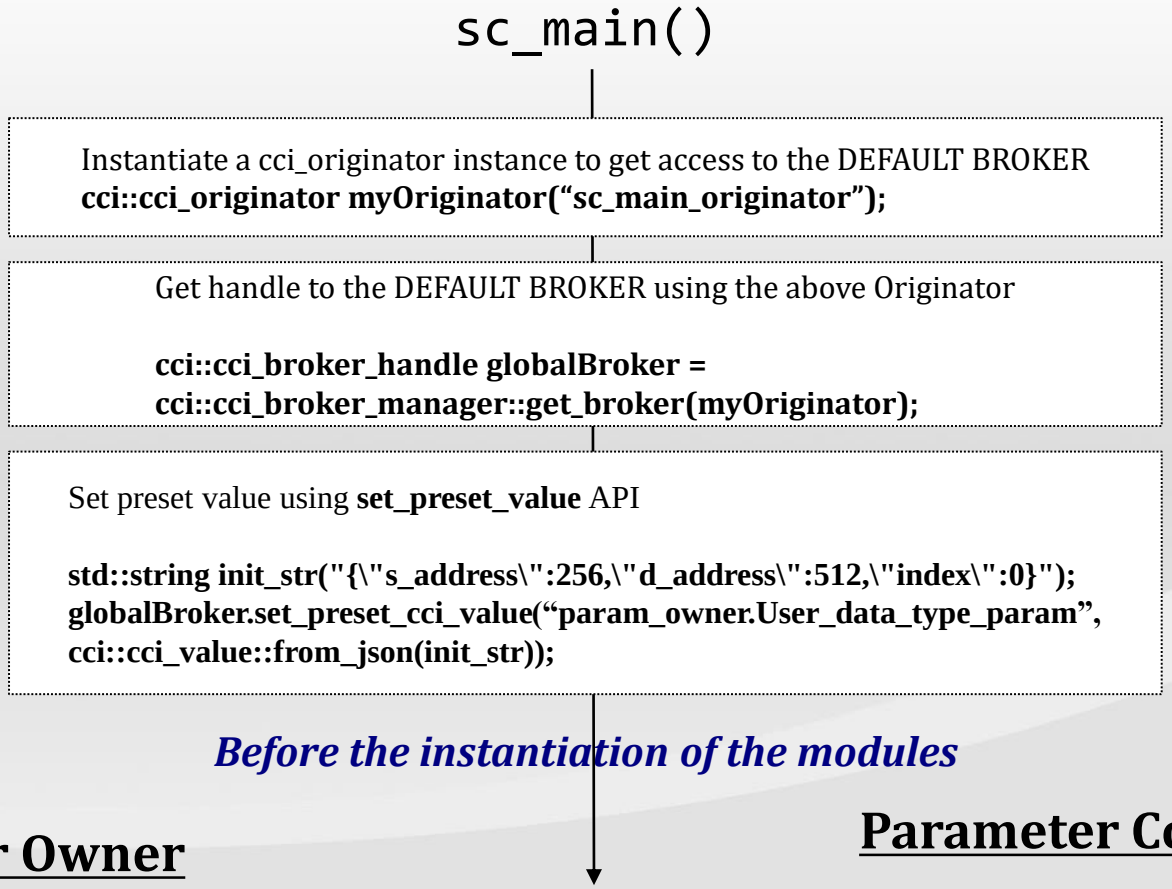
```
struct route_table_ut
{
  // Default Constructor
  route_table_ut()
  : s_address(0x0)    // Source Address
    , d_address(0x0)  // Destination Address
    , index(0x0)     // Index
  {
    // Nothing to implement
  }

  // Overloaded Constructor
  route_table_ut(int saddr, int daddr, int idx)
  : s_address(saddr)
    , d_address(daddr)
    , index(idx)
  {
    // Nothing to implement
  }

  int s_address; // Slave Address
  int d_address; // Destination Address
  int index;     // Index

}; // struct route_table_ut
```

User Defined Data Types

Overloading *insertion* operator of C++

```
std::ostream& operator <<(std::ostream& os, const route_table_ut& ud)
{
  cci::cci_value udv(ud);
  return os << udv;
}
```

User Defined Data Types



User Defined Data Types

Parameter Owner

cci::cci_param<route_table_ut> udt_param;

Set name and assign default value:
udt_param("User_data_type_param",route_table_ut(0x200,0x300,1)

Prior to 0ns

Query parameter name:
udt_param.name();
Returns Name : *param_owner.User_data_type_param*

Query parameter's **Default Value** using **get_default_value**() API:
udt_param.get_default_value();
Returns Value = *{"s_address":512,"d_address":768,"index":1}*

Query parameter's **Value** using **get**() API:
udt_param.get();
Returns Value = *{"s_address":256,"d_address":512,"index":0}*

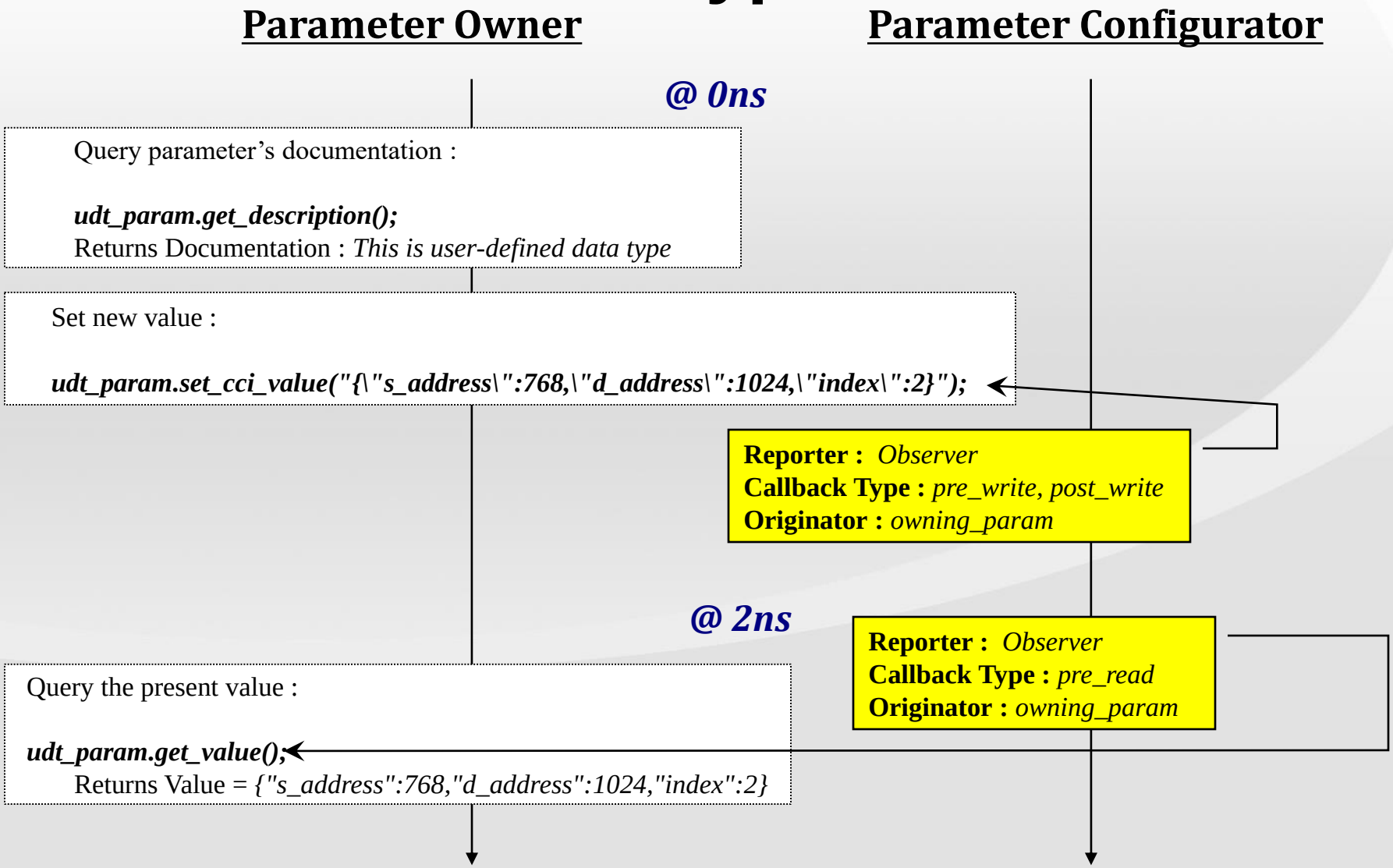
Set documentation :
const std::string init_desc = "This is user-defined data type";
udt_param.set_description(init_desc);

Parameter Configurator

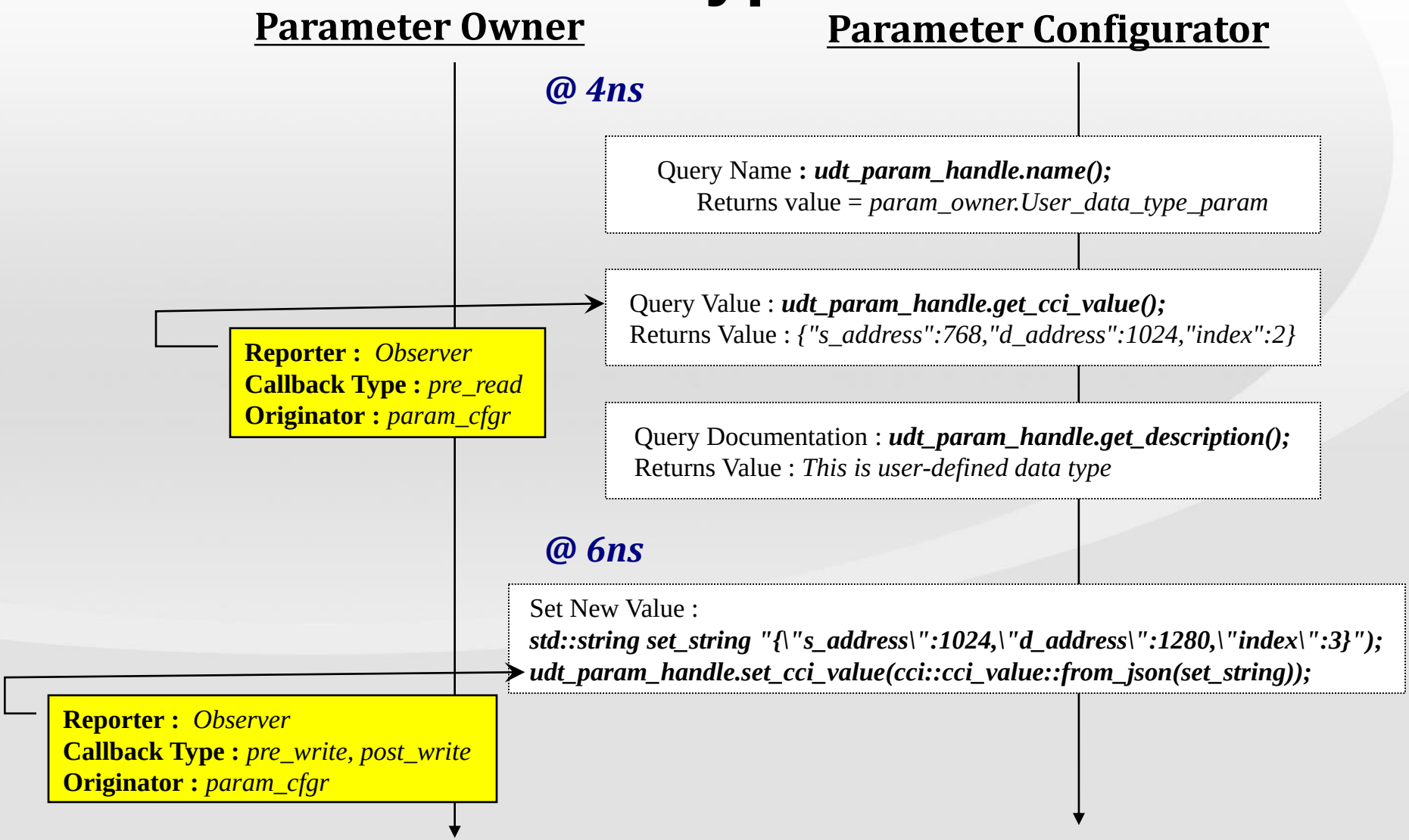
Declare handle for the parameter

cci::cci_param_handle udt_param_handle;

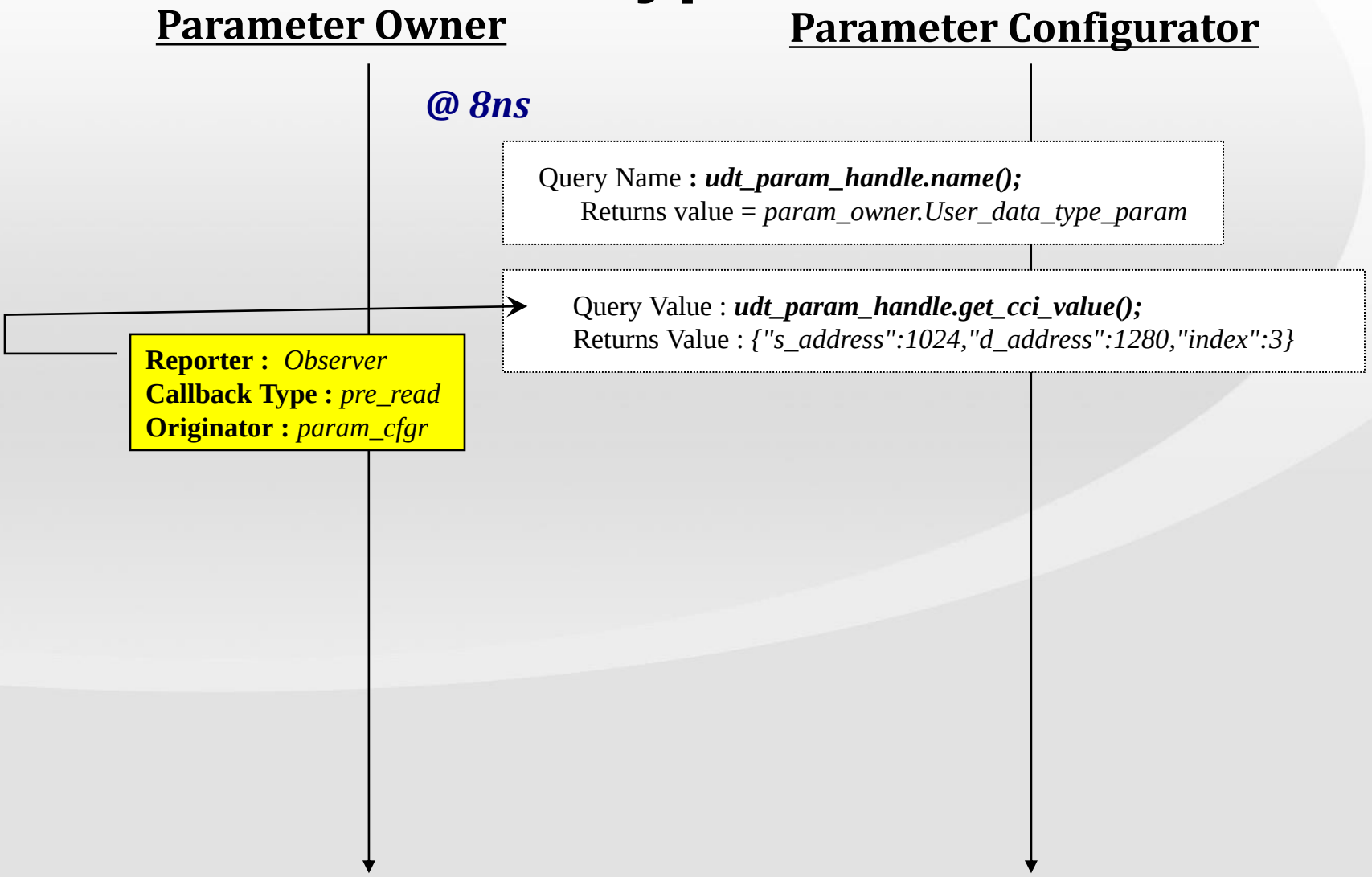
User Defined Data Types



User Defined Data Types



User Defined Data Types



Expected Output
(ex16_User_Defined_Data_Type.log)

```
SystemC Simulation
Info: sc_main: [MAIN] : Setting preset value 's_address:256,d_address:512,index:0' to UDT
Info: param_owner: @0 s, Prior to 0 s      demonstrating 'get_default_value()'
Info: param_owner: @0 s, [OWNER -> Retrieve] : Parameter name  : param_owner.User_data_type_param
Info: param_owner: @0 s, [OWNER -> Retrieve] : Using 'get_default_value()' :
{"s_address":512,"d_address":768,"index":1}
Info: param_owner: @0 s, [OWNER -> Retrieve] : Parameter Value' :
{"s_address":256,"d_address":512,"index":0}
Info: param_owner: @0 s, [OWNER -> Set] : Param desc - 'This is user-defined data type
Info: param_cfgr: @0 s, [CFGR C_TOR] : Broker Type : DEFAULT_BROKER - is not a private broker.
          [OBSERVER C_TOR] : Broker Type : DEFAULT_BROKER - is not a private broker.
Info: sc_main: Begin Simulation.
Info: param_owner: @0 s, @ 0 s
Info: param_owner: @0 s, [OWNER -> Retrieve] : Description : This is user-defined data type
Info: param_owner: @0 s, [OWNER -> Set] : New Value param_owner.User_data_type_param
          's_address:768,d_address:1024,index:2'
          [OBSERVER pre_write_cb] :  Parameter Name : param_owner.User_data_type_paramOriginator info
: param_owner
```

Cont'd

```
          [OBSERVER post_write_cb] :  Parameter Name : param_owner.User_data_type_param
          Originator info : param_owner

Info: param_owner: @2 ns, @ 2 ns

          [OBSERVER pre_read_cb] :  Parameter Name : param_owner.User_data_type_param Originator info
: param_owner

          [OBSERVER post_read_cb]:  Parameter Name : param_owner.User_data_type_param Originator info
: param_owner

Info: param_owner: @2 ns, [OWNER -> Retrieve] : UDT Value :
{"s_address":768,"d_address":1024,"index":2}

Info: param_cfgr: @4 ns, @ 4 ns

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter name : param_owner.User_data_type_param

          [OBSERVER pre_read_cb] :  Parameter Name : param_owner.User_data_type_param Originator info
: param_cfgr

          [OBSERVER post_read_cb]:  Parameter Name : param_owner.User_data_type_param Originator info
: param_cfgr

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter value:
{"s_address":768,"d_address":1024,"index":2}

Info: param_cfgr: @4 ns, [CFGR -> Retrieve] : Parameter desc: This is user-defined data type
```

Cont’d

```
Info: param_cfgr: @6 ns, @ 6 ns

Info: param_cfgr: @6 ns, [CFGR -> Set] : Value - 's_address:1024,d_address:1280,index:3

          [OBSERVER pre_write_cb] :  Parameter Name : param_owner.User_data_type_paramOriginator info
: param_cfgr

          [OBSERVER post_write_cb] :  Parameter Name : param_owner.User_data_type_param
          Originator info : param_cfgr

Info: param_cfgr: @8 ns, @ 8 ns

Info: param_cfgr: @8 ns, [CFGR -> Retrieve] : Parameter name : param_owner.User_data_type_param

          [OBSERVER pre_read_cb] :  Parameter Name : param_owner.User_data_type_param Originator info
: param_cfgr

          [OBSERVER post_read_cb]:  Parameter Name : param_owner.User_data_type_param Originator info
: param_cfgr

Info: param_cfgr: @8 ns, [CFGR -> Retrieve] : Parameter value:
{"s_address":1024,"d_address":1280,"index":3}

Info: sc_main: End Simulation.
```